The sustainability of free/open source software

Giorgio Padrin, Christian Genthon, Fabio Arcangeli

The Sustainability of Free/Open Source Software

Giorgio Padrin (*)
Christian Genthon (**)
Fabio Arcangeli (*)

(*) Università di Verona
(**) Université de Grenoble

Reference address: fabio.arcangeli@univr.it

Paper proposed to the Third EPIP Conference
"What motivates inventors to invent?"
S.S. Sant'Anna, Pisa, April 2-3, 2004

Preliminary draft, February 2004

KEYWORDS:
Economics of Internet, Economics of Software, Free/Open Source Software, Product Life Cycle

JEL: O31, L86

ABSTRACT

The paper applies three analytical frames to a better understanding of the itch to invent and innovate cooperatively, a still inadequately treated stylised fact, while drawing some lessons from an ongoing Free/Open Source software project on communication standards, software and services: **Jabber**, taken as an eloquent case and test bed for the proposed three-layered frame.

The first frame derives form the **territorial innovation systems** literature: some features of the Internet economy, and particularly such standard-setting institutions as IETF working groups, provide a favourable climate to the governance of cooperative software projects.

The second one is drawn from the economic **theory of networks:** the actual inducements to cooperate can be explained by a class of models about the incentives and costs faced by an agent, rationally deciding whether to join a network and betting upon choosing a fitter one.

The third one improves the latter, by introducing a simple evolutionary frame: the **software project lifecycle.**

On the analytical level, a major finding is that economic models overestimate "cooperation failures": if developers were strictly "rational", they should cooperate at a much lower scale compared to observed patterns. This puzzle leads to the suggestion of re-introducing Smithian Moral Sentiments into economic analysis.

As another major point unveiled from the evidence of the case is the sensitive insuppressible key role of **intrinsic motivations** in this kind of innovative enterprises, **linked strictly with the core nature of free/open source style of organizing.** It stems that, in terms of institutional arrangements, there's a wide spectrum of possibilities to experiment with, taking absolutly care not to destroy the vitality of the free ecology mining the critical drives of the innovators.

As far as policies are concerned, the paper aims to switch our attention to the long term sustainability of the novel software-services business models, and a "just" distribution of collective innovations net benefits.

Credits: a little note about the authors more specific contributions to the whole common ecology of this paper:

Giorgio Padrin: technological issues, the Jabber case and the socio-economics of free/open source software;

Christian Genthon: industrial analisys of free/open source issues in the software and information service industries;

Fabio Arcangeli: economic and philosophical frameworks from regional sciences, institutional and technical change economics.

L' homme voudrait être égoiste et ne peut pas.
C'est le caractère le plus frappant de sa misère
et la source de sa grandeur. (S. Weil, 1997, p.388)

1. Introduction

Starting from a post-Schumpeterian approach, the paper aims to add some fresh lines of argument to the current discussion and wide concern, among developers, Free/Open Source software[1] community leaders, technology scholars and policy makers in the search of new solutions to the dilemma: how to preserve the creative and high technical quality of this type of software, while at the same time identifying a more robust business model. In such a way as to increase the economic payoff to developers-and-users, without destabilising its peculiar ecology, that has shown to be able to release high quality software to potential end users. For the sake of finding out escape ways from the medium-long term risks of economic stagnation of Free/Open Source software: the new, fascinating socio-economic sector in search of a sustainable business model.
In the light of the proposed framework, the paper analyses a Free/Open Source software development project: Jabber. The case study is summarised here with a view to the key issues of our post-schumpeterian model:

- High technical quality requirements, namely including the adherence to, and promotion of open standards – that is of interoperability in computer and communication environments;
- factors of the diffusion-evolutionary fitness of a software project, with an aim to attract at the same time both creative software producers and perspective users, and involve the latter since from early project stages;
- the rationale behind their coordination and governance solutions, aiming to find a "social contract" across different agents, namely promoters and users-developers;
- and finally an evaluation of the attempted solutions to "sustainability" : how to increase the resource transfer upwards along the value chain, from the market to basic design and programming tasks.

[1] We propose "free/open source software", in order to encompass all the different components of what has become a large movement, often named nowadays open source software or open sources, and to recall that it was born with the Free Software Foundation.

The objective functions of the core group of programmers and of the other actors of the alliance behind a project, will be some mix (with heterogeneity across converging interests) of technical quality and atmosphere worth participating to the venture (short run net benefits), expected project duration and success or "picking the winner" (long run ones).

But a question immediately arises: which kind of success, measured in which ones of the following, often incompatible dimensions (see the frank and revealing interview by Torvalds 1998)[2].

- The psychological satisfaction by users-developers of their passion for the art-hobby of free and good quality programming?
- The reputation and fame capital of the Author and indirectly of core co-developers, in the professional or even a wider arena? In this case, which and where are the expected fruits of such reputation capital (which: software or other gifts, dollars or euros; where: internal to the cooperating club and digital network; external: social and commercial world)? Or will this fame capital be frozen within digital environments (as Torvalds interviewer was arguing) ?
- The sociological satisfaction - in the light of Aristotle and San Tommaso d'Aquino – accessible even to every baseline co-developer, and drawn by élite-and-mass (at the same time) club membership: participating to small-to-large scale barter trades across cooperating developers, and being part of the "programming intelligentsia" (Bezroukov 2004)?
- Which career perspectives, in which market or non-market segment of the software and services trades, and therefore how, how much remunerated and by whom?
- Or the profitable commercial exploitation of product derivatives and complementary services?[3]

The main tenet of the paper is in parallel (only with a different theoretical location) with Lerner and Tirole's one within mainstream economics: the wide area of the Free/0pen

[2] Crowston et al. (2003) discuss the multi-dimensionality of success of a Free/Open Source project. The quoted interview by First Monday terminates with this exchange:
"*Linus Torvalds:* I really don't think you need all that much "quid pro quo" in programming – most of the good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.
First Monday: Yes, so this means that they see it as a form of self-expression, not as production – play, not work . . .
Linus Torvalds: Yes. Kind of the way artists tend to work: artists usually don't make all that much money, and they often keep their artistic hobby despite the money rather than due to it."

[3] The motivational debate is continuously enriched by new empirical findings: e.g. Hertel et al. 2003, Lakhani and Wolf 2003.

Source Republic of un-commercial software can easily be interpreted, without major changes, within the usual framework of post-schumpeterian theory of invention-innovation-diffusion (Freeman 1988; Arcangeli 1991). The major difference from the usual pattern is in quantity not in quality: in the case of Free/Open Source software only a part, often a minority of the transactions go to the market, while usually the opposite is true (only a small fraction of transactions are held and closed within cooperative alliances - by a combination of barter and money, informal and formal exchanges, knowledge and personnel flows).

As one of the Authors has already noted in a previous contribution, **the sustenaibility of the Free/Open software Republic in the current economic environment is an open question** (Genthon 2004)[4], although there's no fear of collapsing tomorrow for a number of reasons: from the artistic drive of developers, to the direct and undirect support from public and private organizations, and also the strategic interest and investments of many players in the IT industry.

Now the lack of a sound economic model for the transfer of resources from those projects closer to the market, to the basic ones where cooperation prevails and/or is more intense and pure, requires an evolution of the Free/Open Source software constellation, which should reach a new phase as a social system of cooperation, its coordination and governance.

The paper proposes a joint application of three basic analytical frameworks for an interpretation of the itch to invent and innovate cooperatively in software: innovation systems effects; network and cooperation theory; post-schumpeterian views of self-organising and path-dependent processes of swarming around an innovation and its diffusion. By encompassing the entire invention-to-diffusion cycle, the paper supplies an overview of selected issues, and does not specialise on a specific issue of the ongoing debates on Free/Open Source software, such as licensing, developers motivation or the business model.

The first frame is drawn form the **territorial innovation systems** literature (TIS, industrial districts and alike): the key argument is that a set of percolating local environments, with distinctive institutional pillars and cultural landscapes, has been nowadays firmly established in the “Small W W World”. Here, some features of the Internet economy and particularly the standard-setting institutions (IETF working groups) provide a favourable climate to the governance of cooperative software projects;

[4] Genthon (2004) shows that the business model currently suggested and tried by Free/Open Source project leaders is not sustainable, since it is hampered by stronger countervailing forces stemming from the economic *régimes* in the industries of software and information services.

therefore the Digital Commons (like physical ones) are no virgin lands, but social spaces constructed according to repeated interactions between the State, the civil society and private actors.

The second one is drawn from the economic **theory of networks:** as Lerner and Tirole (2002) have shown, in a statically comparative frame the inducement to cooperate is essentially an outcome of incentives and costs for an agent/organisation choosing to join a network.

The third one introduces an evolutionary frame into the latter, with a view to a specific application of a wider model of cooperation to the software industry. This approach stems from a switch of attention from just explaining Free/Open Source software invention- creation, to a discussion of its long term sustainability in such two (currently) distinct, segmented markets as software packages and services (Genthon 2002, 2004).

Software life cycle models, according to this switch of focus, have moved form the software engineering main attention to development phases (the waterfall model, which included by the way maintenance as well), to a life cycle focussing upon co- developers and users swarming in the diffusion phase, that we will discuss in the next Section. In this view, the governance solutions for the coordination of developers' efforts into a single project (therefore also the dynamically optimal licensing tools) should change along a project life cycle, in order to favour and maximise, *ceteris paribus,* the chances of those **swarming processes,** that are actually the drivers of a cooperative innovation and its diffusion.

But the analytical and policy issue is still open and it is left unsolved by such proposals as to move from GPLed products in the early stages to BSDish licensing[5] tools, when the product- project matures: do we accept to move down along a given trade- off curve (if it ever existed) of loosing "developers freedom" and capabilities (therefore increasing the

[5] Let us briefly recall that the GPL is largely considered - even by dissenters – as the beginning of the whole story: a legal and perhaps the major innovation by Richard Stallman and FSF, alongside with his excellent, long hacker programmer activity (Williams 2002); BSD is a prototype of many other "smoother" and more business- oriented licensing agreements promoted by the "Open Source" movement initiated by Eric Raymond (1998, 1999).
Stallman objects to him that a cooperative production model must be preserved even when going toward the market: "C'est ici le risque du movement 'open source', qui ne juge les logiciels que d'après des critères techniques et qui met en avant les bienfaits pratiques de l'ouverture des logiciels. Il est tentant pour des utilisateurs d'évaluer un programme uniquement par ses fonctionnalités et de faire passer au deuxième plan les questions de liberté.
Une des missions du movement 'logiciel libre' est de garder à l'esprit et de rappeler l'importance de la question de la liberté" (Stallman 2001). The close similarity of this thesis with Amartya Sen's life key policy message is obvious, although the two personalities and scholarships differ so much. In fact the similarities go even deeper, since Sen's capability theory is a powerful tool for evaluating the welfare effects of free software.

risk of collapse for the entire "Republic of Science" ecology), in order to attract more swarming waves from mass markets (Stallman 2001)? Or is it possible to devise sustainable ways of coupling collective invention with mass diffusion?[6]

In the next Section we will introduce the main lines of a post-Schumpeterian interpretation of fitness and governance of Free/Open Source projects, by shortly comparing it with some alternative economic models, proposed either by economists and/or by the community leaders. In Section 3 lessons are drawn from a contemporary project on communication software: **Jabber**, a test bed for the proposed frame.

2. The "Programming Intelligentsia" problem: joining the fitter network

This Section is discussing some interpretive paradigms to be applied to the observed stylised facts.

Let us start with the "sympathy/civic sense" couple of Adam Smith: while the Free/Open Source galaxy meets growing consensus and sympathy, across the public opinion and big players, civic sense and fairness should bring us to improve the sustainability of such a collective way to provide some of the necessary bases and tools to a knowledge society. Moreover, it is also of Moral Sentiments we are talking about when referring to cooperation (Smith 1790, Sen 1999)[7]. Not pure altruism nor obedience matter as much, but a superior sense of order and justice, commanding to paying service to the larger system you are embedded into, with priority to, and before concentrating on your closer, local system.

With a surprising sense of modernity before Hegel, who was in fact so much hit by his writings, Smith (1790) uses as a *leit motiv* exactly the couple lower/higher system for a large class of moral choice problems. From the Modernity viewpoint, this is egoism at a more rational, wider looking and long-term scale: the one of the bourgeois "prudent man" where ethics and economics superimpose themselves, although without any mess or convergence across the two independent spheres (Zanini 1991).

[6] If the above named trade-off "developers freedom – mass diffusion" ever existed (an assumption that we do not take here for granted), increasing the sustainability of Free/Open Source software would be a problem of moving such a hypothetical trade-off curve more far away from the origin, or devising business models allowing for both developers idiosyncrasies and mass market demands.

[7] Amartya Sen openly recognizes his important intellectual debts towards Smith, as well as his wife, Emma Rotschild (see e.g. Rotschild 1992), for reinforcing this link.

Amartya Sen (1999, ch. 11) underlines the contemporary analytical implications of a well known important dimension of the Smithian sentiments theory, i.e. the couple sympathy/engagement (or also, as a second term: humanity, generosity, civic sense: Smith 1790, Part IV ch.2); in Sen's view, you don't go very far by just manipulating a standard utility function so as to allow for sympathy (the payoff of j entering the objective function of i)[8].

Within this narrow approach, you miss the non-egoistic centre: exactly the upper side of rationality or the sense of the duties/rights duality in a social contract (rights being what you receive from other people being just to you: Weil 1962), and the true sense of **justice** finally. The latter, rephrased in Emmanuel Levinas words, is the necessity of a third party playing a role between i and j, for the sake of justice in any case: even in order to equilibrate asymmetric or reciprocal excesses of altruism, as well as for the usual staff of opportunism and free riding; something rather similar to what Adam Smith identified with the Social Ego or the Spectator.

Internet: the territory of the free software community

In the light of an integrated economic-and-social view of development and welfare (Smith-Sen), we will now introduce a box of basic tools to be applied to Free/Open Source software:

- lessons from territorial innovation systems for the Web socio-economic environment;
- joining a software project by a potential co-developer as an entry choice into a network;
- a post-Schumpeterian view of the software project life cycle.

As for the first point, a software project is endowed with internal and external institutional infrastructures: on the one hand the social contract across different agents, and the consequent choices about finance, governance and licensing; on the other hand the legal, market and technical frame of Internet and the digital industries.

As for the internal infrastructure, the major players are:

1. the original project Author, core collaborators, other co-developers, the developers of derivative products: most of them belonging to the "programmers intelligentsia" but divided, among other heterogeneities, between full members

[8] Sen names Becker (1998) as a representative of the extensions of mainstream ego-centrism to Smithian "sympathy" for the Other; some recent contributions on related subjects are in Sacco and Zamagni, 2002.

and commercial developers, i.e. part-time members of the free/open Republic (Von Hippel and Von Krogh, 2003);

2. the distributors and service providers, the end users.

Is it possible, as the schismatic "Open Source" movement since 1998 was supposed to try (by promoting BSD-like licensing vs. GPL), a fine tuning of governance tools (the coordination of cooperation), such as to enable to preserve the "Republic of Science" characters of openness, transparency and technical quality; while at the same time creating such science-market bridges as to drive benefits upward the value chain, from the end user market to basic research (the collective appropriability issue we will discuss later on, in the post-Schumpeterian model)?

The coming to an adult age of the Free/Open Source software movement found the Internet social environment as a "necessary but not sufficient condition" for a transition from commercial to cooperative social contracts, i.e. From the cathedral to the bazaar (Raymond 1998). On the one hand, the hacker culture (born in the era of "Unix guys" and groups like the MIT AI Lab one: Levy 1984, Williams 2002, Raymond 2003) became a fundamental, seminal and core component of the new global net culture (Castells 2001, ch.2). On the other hand, the new cooperatively produced software has contributed significantly to create the tools now in use on the Net.

Now the problem is: will "liberated" interaction systems survive the interactions with oligopolistic forces and monopoly elements shaping the external infrastructure of digital projects and industries? And at which social welfare conditions, namely for the large number of end users and their freedom as well? Currently, provided that the anti-Microsoft section of the oligopoly will continue to support the open sourcing (perhaps Microsoft as well?), it will nonetheless be less happy with a survival of a free Republic of Science backing it, unless a division of labour and a value chain favourable to the oligopoly is established.

From the fields of Technical Evolution, Territorial and National Innovation Systems (TIS & NIS), we derive that:

1. the Internet economics and institutions (namely the democratic-technocratic balanced way to select candidate Internet standards by the IETF working groups: Padrin 1996) provide a social and (although with imperfections, covered up in the past by the New Economy bubble) an economic environment for the breeding of software innovations, such as to increase their chances of survival and diffusion above critical mass.

2. Digital meeting places have created “plazas” where fads and word-of-mouth spread faster.
3. Large multiplier effects in software swarming processes, fostering innovation with better diffusion prospects, happened in the Internet socio-technical environments: a giant barter trade (the real Bazaar ...) started up, between hacker communities enlarging to a Programming Intelligentsia, and the WWW.

A network view of Lerner and Tirole’s contribution

As for the second point, let us note that the Lerner-Tirole (2002) taxonomy of the inducements to develop free software can be smoothly re-phrased as a special case of the choice whether to join a network and which one, among alternatives. So re-interpreted, their model might even acquire a dynamic and path-dependent component, as we will see at the end of this Section.

Choice at time t of agent i to join network j or k is based upon the sign of the inequality:

$$B_{ijt} - C_{ijt} + ?_t{}^T A_{ij?} ?^? \quad ? \quad B_{ikt} - C_{ikt} + ?_t{}^T A_{ik?} ?^?$$

Where:

- B and C are respectively:
  - current short term benefits, including developer’s ones as a user;
  - networking and opportunity current costs from cooperative activity;
- A are expectations upon delayed net benefits, i.e. resources flowing from users to developers, in some form or another (gifts, money, etc.).

Please note that if agent i is representing himself as an artist (as suggested by Torvalds 1998), his opportunity cost C of spending time developing might be negative (- C = positive value of fun, if we ignore networking costs).

Moreover, even the hidden talent signalling and career prospects properly stressed by Lerner and Tirole, are somehow affected by Moral Sentiments and the shared values based interactions in a social environment (we refer here mainly to Becattini and Brusco theories of the Industrial District as a rich social interaction environment). Therefore a classical economic-and-social view might be preferable to a *homo oeconomicus* one.

Then the usual *caveats* apply about the emergence of expectations or self-fulfilling prophecies in an environment prone to fads and swarming. What affects the chance of a project to enter path-dependent chains of swarming, passing by from the original

Author, with continuous risks of premature decadence, forking or hijacking, down to the end user? In the current cultural climate in a developers communities, this path-dependent swarming process might continue if the project is really good and interesting, if moral conditions of mutual trust and no commercial hidden scope are clearly stated and believed by potential co-developers, and if no negative marketing gives "bad marks" to the project.

This means that the promoting group must send the appropriate **collective signalling** to the reference community: not so much in terms of communication and marketing ("Show me the code!"), but first of all through its technical, institutional and legal choices in tune with the community (see the instructive discussion of the Jabber case in the Section to follow). In such a way as to impulse and maintain a trust and shared values capital, that creates potential economic value through strong dynamic attractors of agglomeration around a trustable project proposal (by actors with different roles, from the leader to the high end user).

These considerations lay behind the amazingly large and unexplained diffusion of such a radical legal innovation as **free software licensing** (like GPL and BSD-like ones): it must be a sort of DOC label for trustable projects. Once this recognised, one has also to add that the FSF leadership, also due to its cultural, professional and technical roots, might underestimate the sustainability and business model issues.

On the other hand, even when social agglomeration works, we still have an unsolved puzzle for economists: a major obstacle to applying a purely economic motivation model to project participation choice, is that in absence of a working business model for cooperative basic software products, the A flow in the above algorithm will be much below its full economic potential.

But in this case, by being rational, the productive agents (Author and even more co-developers) should anticipate this, under-invest in cooperation, and choose to join a project less often than what we observe. Moreover, "A" quasi-monetary or monetary rewards are highly asymmetrically distributed: which actual career prospects has a baseline co-developer, even if she/he is participating to one or two of the most successful Free/Open Source software projects?[9]

[9] A list of some alive, important ongoing Free/Open Sources software projects (quite unrepresentative of the younger ones) includes: Apache, Bind, Free/Net/Open BSD, GCC, Gnome, KDE, Linux, Perl, PostgreSQL, Ruby, Samba, Sendmail, TCL, TeX. About 12.000 active Free/Open Sources software projects are estimated (Ghosh and Prakash, 2001; Bonaccorsi and Rossi 2003).

These considerations lead us to propose a further amendment to a basic network model, in a post-Schumpeterian economic frame, besides having argued for the relevance of sentiments and trust in social agglomeration processes.

The software project cycle

The Schumpeterian tenet here is that there must be some qualitative coherence (even if a dynamic unbalance of magnitudes might appear) between innovation and its finance (Fumagalli 1995):

- the collective innovation-diffusion process of commodity creation-production-circulation;
- and its dual: a coherent appropriability *régime* in the monetary and credit domain.

In the software domain something similar happens to the material and manufacturing ones: being first to innovate and learn constitutes the best dynamic protection of an invention, much stronger than any legal and licensing one. We propose here a simple key to reading the evolutionary nature of software innovations.

Let us stylise a collective software project cycle (along the classical Vernon-Hirsch approach, applied to our field) of: Infancy – Bifurcation 1 (Bandwagon or early stagnation) – Maturity [10] – Equilibrium – Decline – Bifurcation 2 (De-maturation or death). Here is a brief characterisation of phases in a Vernon-Hirsch style, although we do not draw here their full implications, e.g. on processes geographical location, before any deep inquiry on the subject. It would be interesting also to study why proprietary projects do not seem to follow any similar cycle, as if they were more artificial in nature.

1. INFANCY, usually the first two or three milestone versions; **first swarming** or growth age:
    a. Producers: Author ? Core group ? early co-developers
    b. Users: product unstable, early high-end users enter, take-off takes place
    c. Producer-user interaction (von Hippel): mainly in closed form, with a developer-user
2. EARLY BIFURCATION (sometimes in the increasing user base period):
    a. Bandwagon or **second swarming**, diffusion accelerates:

[10] Please note that, in order to adapt Vernon-Hirsch terminology to the information technology field, in our frame the "mature" software product phase corresponds exactly to the "take-off" one in the standard model.

i. Producers: early developers ? co-developers; they increase until a max.; start up of derivative products
ii. Users: max. contagion, diffusion even outside the programming intelligentsia
iii. Producer-user interaction extends to non-programming end user

b. Early stagnation

3. MATURITY: deceleration of the second swarming processes, as above
4. EQUILIBRIUM:
   a. Producers: being attracted elsewhere, co-developers shrink toward a min. threshold
   b. Users: penetration rate approaches an asymptote or, more often, reaches a max.
   c. Producer-user interaction almost disappears
5. DECLINE or late stagnation: co-developers below min. threshold, project disbands
6. LATE BIFURCATION (stable or decreasing user base period):
   a. De-maturation: an event might bring new agglomeration of developers
   b. Death.

In these scheme Jabber would be located in the second swarming phase (2a).

3. Evidence from the Jabber case

Jabber is an interesting and diversified case study from our perspective and in itself. It deals with many core aspects of the organisation of technology evolution in a net communication society. In this section we give a summary picture of it stressing the motivational issues and the organizational and institutional forms used as supporting social tools.

The technology space

Jabber is at a first view an instant messaging (IM) and presence technology. It is a free software object and trajectory, in a space populated by closed proprietary solutions, pushed by such big corporations as America On Line (AOL), Yahoo and Microsoft, leveraging on their prominent market positions on respectively Internet access, web portals and operating systems. The key for understanding this industry segment from a socio-economic perspective is not software production and supply, but communication

services and network building. The value for users lies in the access to a network by other people to communicate with. The three contenders above strive to collect the widest users base and lock them in.

Jabber is founded around the idea of interoperability and the XML technology. The latter is the well known new pervasive data interchange standards family from the WWW Consortium (the standardisation body for WWW related technologies). At the core Jabber is a router switching in real time XML messages from and to a constellation of different entities: some of these are people using graphical software interfaces (clients), some other ones are various services such as contacts roster management and discovery of other peoples online, other ones are gateways (called transports) taking care of translating and connecting transparently to other IM networks, or any interface to other applications. Moreover, the topology of the network system is not centralised around a unique world server, as in the proprietary solutions, but it is distributed among the variety of personal or group servers, as in the email system.

Thanks to its design, Jabber is more, much more than just an IM. Not only it could possibly integrate email, but as the Internet is moving forward from a pretty static WWW to an interactive galaxy of software and human agents enabling new advanced services and contents co-production, this project is located right on the highest evolution waves.

Birth and infancy: milieu and actors motivations

Jabber was born in late 1998 as a project by Jeremie Miller, a system administrator from Cascade, Iowa. As he says in an interview published by Linux Magazine:

> "I remember playing with ICQ a couple of months after it hit the streets in 1995 or 1996, and I didn't know anybody on it, so I dropped it. Then about a year later, people I knew started using it. So I fired it up again. I gradually started having more friends and co-workers using it, and I could use it from home and from work. Then I had a friend pop up who was on AIM (AOL Instant Messaging). All of a sudden I realized that this was a completely separate network from ICQ, with separate software. (...) I was really into XML at the time. I believe I wrote the third XML parser ever created. So by early 1997, I realized what XML was going to be able to do and I saw where instant messaging was going. I also saw that there were libraries out there where people had reverse-engineered the AIM protocol, the ICQ protocol, and the Yahoo protocol. So I thought, 'If I take these libraries and define an XML format that they can all dump into - then someone can build a client that understands this one XML format, and it could talk to all these other services.' In the same instant I realized, 'Wait a second, they can

> talk to each other without going to any of the other networks; you could have your own IM system.' "

Since from the start Miller conceived Jabber as a broad scope technology, of which the IM was the initial 'acid test'. The basic syntax of XML was stabilised as a standard recommendation from WWWC in February of the year (XML 1.0).

In January 1999, Miller announced its project on Slashdot, the famous online news resource for the community of computer techies, whose subtitle motto is 'News for nerds. Stuff that matters'. He invited people to join, releasing the software as free. That was the starting point of the Jabber community that collected a diversified spectrum of contributors, from a bunch of core programmers to numerous high-end users offering suggestions and testing.

Here are some quotes from early contributors, answering about their motivations to join.

Thomas Muldowney:

> "I started like so many others, by seeing the original slashdot posting. I was working a lot on libfaim/gtkfaim at the time and really didn't like the AOL network. I was really looking for an open IM solution and that just happened to pop up. I guess I saw the story right when it went up because I was the second person to sign up on the development mailing list after jer [Jeremy Miller]."

Ryan Eatmon:

> "In the fall of the 1999, I was looking for a messaging product that I could customize. I first started with zephyr (which I had used in college) but got bogged down in the code. In desperation I went out to see if there were any other products available, and I found Jabber. My background is in Perl, and at the time there was a small Perl group that was just getting going. I jumped in and started contributing some code while learning about XML and Jabber. Three years later and I'm still here and still working hard to make Jabber a success".

Dave Smith:

> "My first post to the JDEV mailing list was on Jan 12, 1999. I was in college at the time, and over the Christmas break had decided to write an IM system to deal with all these stupid IM clients I had to run ... only to find out that Jer was way ahead of me. So I got 'assimilated' (resistance was quite futile), and the rest is history."

The Jabber community also grew by linking to other communities, grace to people acting as interfaces with complementary projects, such as desktop environments and programming languages.

In year 2000, in the general atmosphere of the New Economy boom, the project attracted the interest of venture capital and Jabber.com (now Jabber, inc) was founded, hiring two

of the developers including Miller, and a few other ones after. This company proposed itself as a commercial reference for such large business clients as Disney and France Telecom in supporting their implementation of services based on Jabber technology. The free software nature of the project stood unchanged, also if some concerns arose in the community from time to time.

The JSF and standardsation

As soon as in 1999, Jeremie Miller asked the community for help in submitting the Jabber protocols to the IETF standards evaluation process. The IETF is the Internet standard body of the large open international community of computers and networks engineers, organised in a distributed form. In 2000 there was a failed attempt to contribute to the already established IETF working group IMPP.

The interest in Jabber was growing and there were many free software projects and commercial entities activities building on it. The documentation effort was lacking and the community needed to consolidate. For these reasons the Jabber Software Foundation was formed in 2001 on behalf of the Jabber community as a no-profit organisation. Its mission is to promote the use of Jabber protocols and to manage the open and documented development of their expanding set. The JSF processes of standardisation were drawn on the blueprint of IETF, based on open revision and general consensus.

JSF has a membership based on cooptation and granted to the most involved people or companies, and two elected bodies: a technical one, the Council, to supervise the standardisation processes and the technology evolution; and one taking cares of business matters. It receives sponsorships by companies building or using Jabber-based technology, including some big commercial players of the computer industry.

The work in the JSF supported also the creation in late 2002 of the working group XMPP of the IETF, and fastened the path that has lead to the approval as Proposed Standards of the two core protocols in January and February 2004, paving the way to other ones.

The running horse: milieu and actors motivations

The year 2002 witnessed the spreading around of the Jabber project in all respects.

First of all from the viewpoint of the community of developers: the work on the new version 2 of the server software based on a new architectural design sprouted its wings, with an enlarged, internationalised core team, sweeping away every remaing concerns relating Jabber,inc role , therefore increasing the trust capital.

This coding effort recently, in February 2004, produced the first stable release suitable for deployment in production environments. The file AUTHORS in the source code acknowledges Jeremie Miller, Ryan Eatmon, Thomas Muldoney, Rob Norris as the four culprits, and a series of other contributors.

The guy charged to supervise for this new development effort is Rob Norris, who worked extensively on the new architecture. Rob Norris is a system administrator from Melbourne, Australia, working at the Monash University, specifically charged of the messaging and calendaring system.

From a motivational and lifecycle point of view, it is interesting to quote his message in March 2002 to the JADMIN mailing list, a list dedicated to Jabber servers administrators around the world. He replied to a message about next step goals that ended with: “But those things take time, energy, intelligence, and dedication. We have a small number of people with all those qualities. We need more. Any suggestions on how to find them?” Norris:

> “I've been following Jabber's progress in the last year, and trying to get involved where I can, and from what I've seen, such people do exist and are willing to work. I think the problem (at least for me) is that we (the development community at large) don't always have a good understanding of what the goals of a particular project are. (...) Sometimes I have had ideas for things to work on, and most of the time, I've written them and got them to some semblance of usefulness. However, if its someone else’s idea and someone else’s project, I'm not going to work on it until I'm sure that I'm working towards the same goals that everyone else is. (...) I've wondered in the past just exactly how much work brainstorming and drawing on whiteboards and such goes on within the halls of Jabber, Inc. To me, it seems that fully-fledged ideas, occasionally half-implemented, come out of there with some regularity. However, when the rest of the community finally get word of what's going on, all the interesting bits are done - its no longer fun! I have the time, energy and intelligence to work on Jabber. Working on Jabber-related projects is all I do in my spare time. However, its hard to be dedicated if you don't feel like you can take some ownership over a project. And again, I'm not saying the core team are trying to run everything. Far from it, I know that they want others involved (we wouldn't be having this discussion if they didn't!). And I want to be involved, but be _really_ involved, and not just standing on the edge. That's what it feels like sometimes. Admittedly, I know I could do more. I'm mostly waiting for enough spare time (only a few weeks until I get some parts of my life back). But I do want to work on jabberd 1.5, at the very least. So tell me what's next, and you've got me on board.”

Some months later he was charged with the responsibility of the development effort on the new server version 2 (which took over the 1.5 effort). Even Eatmon, an Electronic Design Automation developer at Texas Instruments, leapt to involvement at the level of architecture design, after having cooperated earlier to subprojects, mainly linked with the Perl scripting language.

The web resource Jabberstudio helped serving as a hub and focusing device of the efforts for the developers community, supplying also a www interface to the cvs collaboration tool, and easing the collection of contributions from the wider community.

As for Jabber networks diffusion, there was an acceleration and now more than 215.000 servers are estimated. System administrators working in organisations, institutions and commercial companies implement and customise the platform for the messaging and collaboration needs of the users they are serving.

By overviewing the commercial companies building their business on a Jabber base, we can identify mainly some IT consultants and software developers, and communication service companies. The first category mainly offers technical support to client companies in deploying and customising their systems based on Jabber, in some cases implementing upon their commercial software platform comprising servers and clients, for example clients for cellular phones or wireless handhelds. In the second category an interesting case is France Telecom, also an investor in Jabber, inc. France Telecom's project builds on Jabber a service of IM for its cellular network, accessible by the users via SMS and WAP. France Telecom, as a player in the communication services market, has an interest not to loose control on the software enabling its services in favour of proprietary solutions and networks.

In March 2003 an online survey was held and published on the JSF site addressed to server administrators (Server Admin Survey 1.0); the report gives a hint that:

- 70% of the servers were operating since less than one year, confirming the explosion in year 2002;
- in 94% of the cases the main, free software distribution of the server is adopted;
- the first reason for preferring Jabber is its free software nature, followed by extensibility, low cost, interoperability and security.

IPR policies and major lessons from the case study

The Intellectual Property Rights policy adopted since from the infancy or early Jabber project life might be summarised as follows:

- GPL for software. Symmetric nature and self-preservation in time. Robust against free riding from appropriation by privatising agents.
- Creative Commons license for protocols standards.
- Trademark is managed by JSF (even if it is still owned by Jabber, inc.), to license to projects claiming compliance with the Jabber standards.

A flash note on patents. Microsoft has a set of small patents, but AOL has a wide covering patent on IM issued by USPTO (the USA patents office). It's enough a rapid look at it to note that there is no technological content disclosed. The content is all in the economic games dimension, confirming also in the IM case the wide literature analysing the strategic use of patents.

We can now summerise some important facts and lessons from the Jabber case:

- Project started up from user needs (author's dissatisfaction as a user), a technological idea and a creative itch.
- Sensitive insuppressible key role of intrinsic motivations: "fun" and the creative dimension in search for a free space to express.
- Active experimentation of new distributed organizational forms for knowledge and software co-production.
- Jabber fits into an important market niche, with a huge potential for derivative products.
- It links up ecologically with Internet open standards.
- 'Free software' GPL licensing regime was no obstacle to business alliances and commercial exploitation: on the contrary, it provided a trust capital to build upon, reinforced by guarantees and proofs of an autonomous project management.
- Passing over early bifurcations and a critical mass threshold is necessary.

4. Conclusion

As a way of conclusions, we introduce here some typical examples of how, by further improving the state-of-the-art of our understanding of the economic and social dynamics implied by the emergence of the Free/Open Source galaxy, new private, collective and public strategies might be devised. We put this in the form of a

provisional list of findings and connected governance rules: milestones for further research and economic policy debates.

1.1 First stylised fact. A delicate ecology characterises the Programmers Republic of Science. If one wants to release free software (and not proprietary one), produced in a cooperative (not a hetero-directed) way, she/he will have to follow a narrow path, because:

- on the one hand a cooperative project quality is affected by trust issues;
- on the other, she/he has to cope with a variety of market and non-market links, connecting the project to its client base, which is subject to lock-in phenomena in oligopoly-ruled markets.

1.2 First policy rule: avoid to destroy this "free" ecology in order to reach a mass market.

2.1 Second stylised fact. Swarms with positive feedbacks characterise a project life and dynamics; around a core team, different groups of developers are gradually or suddenly agglomerating: co-designers, socially motivated co-developers and finally derivative product developers, very often more economically motivated. At the same time adoption waves enlarge interactions to users. A variety of social contracts and legal solutions have been tried in the Free/Open social laboratory, in order to deal with this relational complexity.

2.2 Second policy rule: on the one hand some degree of project governance flexibility might help to adapt to, and take more advantage from each single swarming phase (see e.g. the institutional changes in the Jabber case). On the other hand the chances of attracting first, then keeping the best resources, are largely based on trust and creativity (vs. traditional professional labour market and organisation rules).

Reducing the impact of, and watering down the "copyleft" radical legal innovation is not likely to help reaching both targets (flexibility and trust).

3.1 Third. We formulate the hypothesis that embedding Free/Open social contracts in a new business and institutional environment, more coherent with the new social demands, might increase the monetary resources and externalities moving upstream the value chain.

3.2 Third policy rule. Suggested menu of answers to the Free/Open sustainability problem:

- direct or indirect public support, at least in the early transition towards sustainability, when a "Visible Hand" might concur to create bridging institutions, when the invisible one fails to;
- a movement of the social forms themselves, emerging from relationships across and within digital communities, towards a self-sustaining digital collective innovation system.

References

Arcangeli, E.F. (1991), *The Dynamics of Diffusion and the Emergence of Computer Integrated Manufacturing (CIM)*. Ph.D. thesis. Brighton: SPRU, University of Sussex.

Bezroukov, N. (2004), 'BSD vs. GPL. Parts 1-3'. Accessible at: http://www.softpanorama.org/

Becker, G. (1998), *L'approccio economico al comportamento umano*. Bologna: Il Mulino.

Bonaccorsi A. and C. Rossi (2003), 'Why Open Source can succeed'. Research Policy, 32 (7): 1243-58.

Castells, M. (2001), *The Internet Galaxy: Reflections on the Internet, Business, and Society*. Oxford: Oxford University Press.

Creative Commons, http://creativecommons.org .

Crowston K., H. Annabi and J. Howison (2003), 'Defining Open Source Software Project Success'. Paper delivered at the 24th International Conference on Information Systems. Available at: http://citeseer.nj.nec.com/cs

Cusumano, M.C. (1992), 'Shifting economies: from craft production to flexible systems and software factories'. Resesarch Policy, 21 (5): 453-480.

Di Bona C., S. Ockman, M. Stone eds. (1999), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly.

Foray, D. and J.B. Zimmermann (2001), L'économie du logiciel libre: organisation coopérative et incitation à l'innovation, Révue Economique, 52: 77-93.

Freeman, C. (1988), Introduction. In: Dosi et al. eds., *Technical Change and Economic Theory*. London: Pinter, pp. 1-8.

Fumagalli, A. (1995), *Moneta e tecnologia*. Milano: Angeli.

Genthon, C. (2002), *Analisi settoriale, metodologia e applicazioni*, Padova: CEDAM.

Genthon, C. (2004), 'Quelle insertion des editeurs de logiciel libre dans le monde marchand?' Terminal, 90, Autome/Hiver.

Ghosh, R. and V.V. Prakash (2001), 'The Orbiten Free Software Survey'. First Monday

GNU General Public License, http://www.fsf.org/licenses/gpl.html .

Hertel, G., S. Niedner and S. Hermann (2003), 'Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel', Research Policy 32 (7): 1159-77.

Lakhani, K. R. and Wolf R.G. (2003), 'Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software'. MIT Sloan School Management WP 4425-03. Available at: http://opensource.mit.edu/papers/lakhaniwolf.pdf

Lerner and Tirole (2002), 'The simple economy of Open Source'. Journal of Industrial Economics, 52: 197-234.

Levy, S. (1984), *Hackers*. New York: Anchor/Doubleday.

Moody, G. (2001), *Rebel Code. Linux and the Open Source Revolution*. Allen Lane.

Padrin, G. (1996), 'Organizzazioni a Web. Laboratorio organizzativo Internet'. Tesi in Ingegneria gestionale, Università di Padova.

Padrin, G. (2001), '*q-spazi modello 1. Organizzazioni adattive distribuite nell'economia della conoscenza in rete*', tesi di dottorato, Università di Padova.

Raymond, E. (1998), 'The Cathedral and the Bazaar', v.1.33, February 16.

Raymond, E. (1999), *The Cathedral and the Bazaar: Musings on Linux and Open source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly.

Raymond, E. (2003), 'A Fan of Freedom: Thoughts on the Biography of RMS.' http://www.catb.org/~esr/writings/rms-bio.html

Rotschild, E. (1992), Adam Smith and Conservative Economics. Economic History Review 45, February.

Sacco P.L. and S. Zamagni eds. (2002), *Complessità relazionale e comportamento economico. Materiali per un nuovo paradigma di razionalità*. Il Mulino, Bologna.

Sen, A. (1999), *Development as Freedom*. New York: Alfred A. Knopf.

Smith, A. (1759, rev. 1790), *The Theory of Moral Sentiments*. Vol. I of the *Glasgow Edition of the Works and Correspondence of Adam Smith*. Oxford, 1976.

Stallman, R. (1999), The GNU Operating System and the Free Software Movement. In: Di Bona et al. (1999), pp. 53-70.

Stallman, R. (2001), 'La vraie opposition est entre le logiciel libre et le logiciel propriétaire', Interview, Terminal 80-81.

Torvalds, L. (1998), 'What motivates free software developers?' Interview, First Monday.

Von Hippel, E. and G. Von Krogh eds. (2003), 'Special Issue: Open Source Software Development'. Research Policy, 32 (7): 1149-1292.

Weil, S. (1949 [1], 1962 [2]), *L'Enracinement*. Paris: Gallimard. Also pp.1025-1218 in S. Weil (1999), Oeuvres – F. de Lussy ed. Paris: Gallimard.

Weil, S. (1997), *OEuvres complétes*, vol. VI, tome 1: Cahiers (1933 – septembre 1941).

Williams, S. (2002), *Free as in Freedom*. Sebastopol, CA: O'Reilly. GFDL licensed and also available at: http://www.faifzilla.org/toc.html

Zanini, A. (1991), 'Adam Smith: filosofia e mercato'. Introduction to: A. Smith, *Teoria dei sentimenti morali*. Roma: Istituto della Enciclopedia Italiana, pp. XIII-LXXXIX (Italian edition of Smith 1790).